

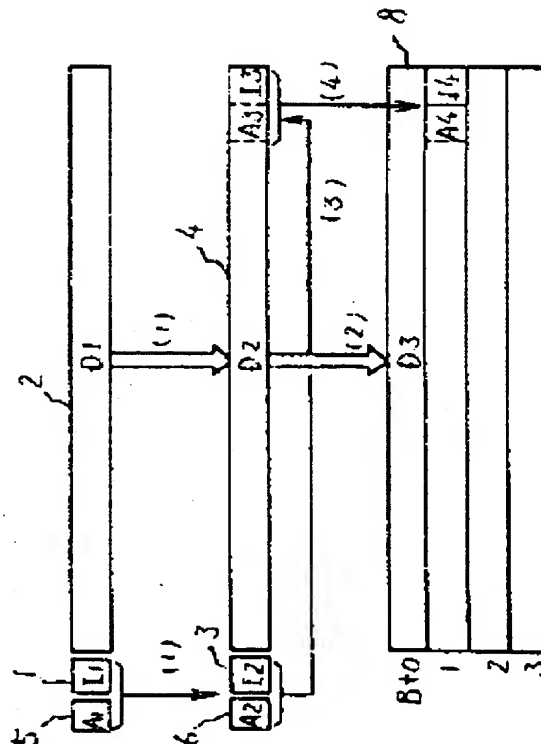
# METHOD FOR MONITORING COMPUTER

**Patent number:** JP57052946  
**Publication date:** 1982-03-29  
**Inventor:** SATOU TOMOKATSU  
**Applicant:** HITACHI LTD  
**Classification:**  
- International: G06F3/14; G06F9/00; G06F9/06; G06F11/30  
- european: G06F9/22D  
**Application number:** JP19800127250 19800916  
**Priority number(s):** JP19800127250 19800916

Report a data error here

## Abstract of JP57052946

**PURPOSE:**To prevent increase in number of program steps, by reflecting operational results of computers in expanded parts which expand a plural number of bits, against the specified length of instruction words handled by computers. **CONSTITUTION:**The content D1 of a memory 2 is transferred to an accumulator 4. At this time, the content of an instruction bit 1 and an operational result A1 in an auxiliary memory 5 are automatically transferred to auxiliary registers 3 and 6. Then, the content D2 of the accumulator 4 is transferred to an address B of a control memory 8. At this time, data transference from the auxiliary registers 3 and 6 to the control memory 8 is inhibited. Moreover, contents of the auxiliary registers 3 and 6 are transferred to the accumulator 4. Then, the content of the accumulator 4 is transferred to an address B+1 of the control memory 8. Through the above mentioned procedure, contents of the memory 2, instruction bit 1 and auxiliary memory 5 are planted in the control memory 8.



Data supplied from the **esp@cenet** database - Worldwide

**THIS PAGE BLANK (USPTO)**

⑨ 日本国特許庁 (JP)

⑩ 特許出願公開

⑫ 公開特許公報 (A)

昭57—52946

⑤ Int. Cl.<sup>3</sup>

G 06 F 9/06

3/14

9/00

11/30

識別記号

1 0 1

庁内整理番号

6745—5 B

2116—5 B

6745—5 B

7368—5 B

⑬ 公開 昭和57年(1982) 3 月 29 日

発明の数 1

審査請求 未請求

(全 7 頁)

⑭ 計算機のモニタ方法

株式会社日立製作所大みか工場  
内

⑮ 特 願 昭55—127250

⑯ 出 願 人 株式会社日立製作所

⑰ 出 願 昭55(1980) 9 月 16 日

東京都千代田区丸の内 1 丁目 5

⑱ 発 明 者 佐藤朝勝

番 1 号

日立市大みか町 5 丁目 2 番 1 号

⑲ 代 理 人 弁理士 高橋明夫

明 細 書

発明の名称 計算機のモニタ方法

特許請求の範囲

1. 一定語長の命令を記憶装置から取込んで順次プログラムを実行する計算機の記憶装置の語長をプロセッサの命令語長に 1 ビット又は複数ビットを加えた拡張語長とし、この拡張語を読出し書込みのできる補助レジスタをプロセッサ側に設け、プロセッサが記憶装置を読出す時は前記拡張語が補助レジスタに自動的に書込まれ、プロセッサが記憶装置にアキュムレータの内容を書込む時はアキュムレータの内容が記憶装置に書込まれると同時に補助レジスタの内容が記憶装置の拡張部分に自動的に書込まれる命令語長拡張方式の計算機において、記憶装置に演算結果を記憶しておく補助メモリを、プロセッサに上記演算結果を自動的に取込む補助レジスタを設けてプログラムをモニタした場合プログラムの内容と演算結果とを同時にモニタ可能としたことを特徴とする計算機のモニタ方法。

(1)

発明の詳細な説明

本発明は計算機のモニタ方法に関し、さらに詳述すれば、一定語長の命令を記憶装置から取込んで順次プログラムを実行する計算機の記憶装置の語長をプロセッサの命令語長に 1 ビット又は複数ビットを加えた拡張ビットとし、この拡張語を読出し書込みのできる補助レジスタをプロセッサ側に設け、プロセッサが記憶装置を読出す時は前記拡張語が補助レジスタに自動的に書込まれ、プロセッサが記憶装置にアキュムレータの内容を書込む時はアキュムレータの内容が記憶装置に書込まれると同時に補助レジスタの内容が記憶装置の拡張部分に自動的に書込まれる命令語長拡張方式の計算機におけるモニタ方法に関するもので、特にモニタ手順を簡略化し記憶装置の使用効率を向上させようとするものである。

計算機はマイクロプロセッサの出現によりデジタル制御の分野に積極的に応用され始めており、これを扱うオペレータの層も拡大されてきている。これに伴い、制御システムへの要求は従来の機能

(2)

重視から操作及び保守性の容易さを重視する方向にあり、このため計算制御システムの操作も、専門技術者ではなく、一般の技術者が行なえるようにすることが要請されている。これは、計算制御システムのプログラム作成、修正、操作を電卓を扱うような簡便さで行ないたいということ、さらにCRTに表示して、より直感的に計算制御システムの動作を把握したいということである。この要請は1ビットの演算機能を有するシーケンサにおいて実現されている。次にこの事例を述べる。

第1図はシーケンスの一例を示す。I/O 0 ~ I/O 3は入力信号を示す。I/O 10は出力信号を示し、入力信号I/O 0 ~ I/O 3の演算結果が出力される。このシーケンスのプログラム例を第2図に示す。演算ステップ1でI/O 0の内容をアキュムレータAに取込み、ステップ2でアキュムレータAの内容とI/O 1の内容との論理積をとり結果をアキュムレータAに格納する。ステップ3でI/O 2の内容をアキュムレータAに取込むと、それまでアキュムレータAに格納され

(3)

りMOに格納しておき、このワーキングメモリ内容により論理記号のフルトン、ハーフトンの区別をする手段が採用されている。

しかしながら、この方法は、シーケンスをCRTに表示でき制御状態が理解しやすい利点はあるが、次の問題点がある。即ち、第4図に示すように第2図と比較してプログラムステップ数が約30%増加することから記憶装置の使用量が増加しコスト上昇になること、及び同一の制御に対して時間がかかることである。

本発明の目的は、以上説明したシーケンスのCRT表示化に伴って発生した、プログラムステップ数の増加を防止することのできる計算機のモータ方法を提供するにある。

本発明の特徴は、計算機が扱う規定の命令語長に対して、複数ビット拡張した拡張部分に計算機の演算結果をも反映させる方法とするにある。

本発明を図面により説明する前に、まず、計算機固有の命令語長に対する拡張語の機能を説明する。命令の拡張語は、記憶装置に格納されている

(5)

特開昭57-52946(2)

ていた内容(A)-1はSTACKと呼ばれるレジスタに自動的に格納される。ステップ4でアキュムレータAの内容とI/O 3の内容との論理積をとり、結果をアキュムレータAに格納する。ステップ5では特殊オア命令(OR)を実行し、アキュムレータAの内容とステップ3で格納されたSTACKの内容との論理和がとられ、結果がアキュムレータAに格納される。ステップ6でアキュムレータAの内容がI/O 10に出力される。以上で入力I/O 0 ~ I/O 3が演算され、結果がI/O 10に出力されることになる。

次にこのシーケンスをCRTに表示する場合、次のような要求が出てくる。即ち、途中の演算結果もCRT上に表示したいということである。例えば次のようである。第1図においてI/O 0とI/O 1との論理演算結果が論理・1・の場合は論理記号1をフルトン(明るく)表示し、論理・0・の場合は論理記号1をハーフトン(暗く)表示する。この要求を満たすために、従来、第3図に示すように、途中の演算結果をワーキングメモ

(4)

内容のみの情報から、命令を解釈し、オペレータが判りやすい記号例えば「LOAD」というように表現しようとして考え出されたものである。

まず第5図を用いてプログラムの内容確認法を考えよう。一般の計算のプログラムは第5図に示されるように、命令をアドレスに従って格納してある。計算機は、この命令に従って命令1、命令2というように命令を実行していくため、次に実行すべき命令のアドレスは現在実行中の命令からわかり順次命令を実行してゆくことができる。しかしながら、計算機に不具合が発生し、第5図に示されるようなプログラムを調べようとしたときは、調査が非常に困難である。このことを第6図により説明しよう。この例は2語命令、即ち2語で1命令を表わす命令である。この命令は(1)に示すようにコアメモリには2進数で書き込まれている。我々にはこの2進数の解釈は困難である。このため(2)に示すようにこれを16進に変換して扱っている。2進から16進への変換は容易に行なうことができるためである。しかしながら、100

(6)

種を超える命令を16進数で扱うのは、専門技術者でなければ理解しにくい。一般技術者が扱う場合は、(3)に示される様に、日常使用している文字で命令が表現される方が理解し易い。次に、コアメモリに格納されている2進数から文字に変換する場合の困難について説明しよう。第5図において、アドレスAの命令1を文字で表現する事を考えよう。A番地が命令の先頭である事を知つていれば次の様に表現が可能である。

A番地の内容(16進表示) F 0 0 1 → LD

A+1番地の内容(16進表示) 0 0 1 0 → 10

ところで、A番地の内容「F 0 0 1」から「LD」と変換するためには、A番地が命令の先頭であることを知つていなければならぬ。それは、命令の先頭は「LD」等文字で表現する必要があり、命令の非先頭は、アドレス又はデータである為、数値で表現する必要があるためである。即ちコアメモリの内容を直接文字を数字で表現するには、コアメモリに命令の先頭と非先頭を区別する情報が必要ということである。これを實現す

(7)

は論理0を書き込むものとする。このようなメモリ構成であると、先に説明したように、なす命令ビットを見て、これが論理1の場合は、我々が理解しやすい文字で命令を表現し、命令ビットが論理0であればこれはデータまたは、アドレスであることを示しているから、数値で表現する。このようにしてメモリ内容が我々に容易に現解できるようにすることができる。なお計算機自体は、命令を順次実行して行けばよいので、命令ビットを参照する必要はない。即ち、現に使用している計算機をそのまま使用してよいわけである。この命令ビットはビットメモリを追加すればよいわけで、計画時に考慮しておけば、命令ビットの追加は容易であり、コストアップもわずかである。

では、この命令ビットへの書き込み、脱出しがどのように行なわれるかを次に説明しよう。第8図において、1はメモリの命令ビット、2は16ビットから成るメモリで、計算機が演算する命令そのものが格納される。3は補助レジスタで、命令ビットに対応し、命令ビットの脱出し、書き込み

(9)

特開昭57- 52946(3)

るには、命令の先頭と非先頭を区別できるように考慮した計算機を使用すればよいことになる。しかしながら、このような計算機は見当らない。この理由は、計算機自体にとつては不要なものだからである。現在実用されている計算機には次のような語長のものがある。すなわち、4, 8, 12, 16, 24, あるいは32ビットを一語としている。たとえば、命令の先頭と非先頭を区別可能な計算機、5, 9, 13, 17, 25, 33ビットを一語とするようなものは実用されていないのである。またこのような非標準の計算機システムをつくることは、開発期間開発費用等の面から不可能に近い。このような状況にもかかわらず、計算機を一般技術者が扱えるようにという要請は極めて大きいものがある。これらの要求を満すための具体策を第7図および、第8図により説明しよう。第7図はメモリのA番地からの内容を示す。このメモリは0~15ビットと、1ビットから構成されている。1ビットは命令ビットと称し、命令の先頭語には論理1を書き込み、命令の非先頭語に

(8)

用いる。4はアキュムレータで、演算結果がここに格納される。10はアキュムレータ4とメモリ2を接続するデータバスである。9はメモリの命令ビット1と補助レジスタ3を接続する専用ラインである。7はデータバス10の16本のうちの1本と接続されているデータバスである。

計算機はメモリ2の内容をアキュムレータ4に転送するか、或いはアキュムレータ4の内容を2に転送するかのいずれかの作業を行つている。この時、補助レジスタ3はあたかもアキュムレータ4の様に動作し、アキュムレータ4からメモリ2にデータを転送するとき、補助レジスタ3から命令ビット1にデータが転送されるように構成する。

逆に、メモリ2からアキュムレータ4にデータが転送されるときには、命令ビット1から補助レジスタ3へもデータが転送されるように構成する。このようにすると、データ転送に関しては、計算機の語長が1ビット拡張されたことが理解される。ただし、データ転送以外の演算において、補

(10)

助レジスタ3はアキュムレータ4の拡張とは見なされない。例えば、アキュムレータの内容をシフト演算しても、補助レジスタ3の内容がアキュムレータ4に移るということはない。これは補助レジスタ3が、既製の計算機のアキュムレータ4と並置しただけのもの故、当然のことである。それではプログラムをチェックするものにとつてこの補助レジスタ3はどのように機能するかについて、さらに検討を進めよう。

計算機は、第9図に示されるような構成が一般的である。それは、演算を実行する処理装置12、プログラムを格納するメモリ2、プログラミングを可能とするプログラマ11、プログラマ11からデータを受け取り、またはプログラマ11へデータを送信するプログラマ制御回路13、プログラマ制御回路を制御するコントロールメモリ8から構成される。この計算機システムは、通常処理装置12、メモリ2で動作しているが、プログラマ11から指令を受け取ると制御の中心はコントロールメモリ8に移り、プログラマから転送され

(11)

アドレスB+1に転送する。以上の手順を経て、メモリ2および命令ビット1の内容をコントロールメモリ8に格納できる。以後は、B番地の内容D3、B+1番地の内容I4からメモリ2の内容を解読できることが理解できよう。なお、I1~I4は同一データで、サフィックス1~4はデータの移動順序を示す。D1~D3も同様である。また(1)~(4)は転送順序を示している。

次に、プログラマから伝送されてきたデータをメモリに書き込む場合について説明する。第11図において、1は命令ビット、2はメモリ、3は補助レジスタ、4はアキュムレータ、8はコントロールメモリである。今、コントロールメモリには、プログラマから伝送されてきたデータが図示の如く、即ちB番地にデータD1、B+1番地に命令ビットデータI1が格納されているものとしよう。第1に、コントロールメモリB+1番地の内容I1をアキュムレータ4に転送する。この時、コントロールメモリ8から補助レジスタ3へのデータ書き込みは禁止される。即ち、補助レジスタ3の

(13)

特開昭57- 52946(4)

て来たデータのメモリ2への格納、またはメモリ2の内容読出し等を実行する。プログラマからの指令に対する実行が完了すると、制御の中心は再びメモリ2に移るのである。次に、補助レジスタ3の動作を第10図、および第11図を用いて説明しよう。始めに、プログラマがメモリの内容を読む場合について説明する。第10図において、1は命令ビット、2はメモリ、3は補助レジスタ、4はアキュムレータ、8はコントロールメモリである。第1にメモリ2の内容D1がアキュムレータ4に転送される。この時、自動的に命令ビット1の内容11が補助レジスタ3に転送される。第2に、アキュムレータ4の内容D2をコントロールメモリ8のアドレスBに転送する。この時、補助レジスタ3からコントロールメモリ8へのデータ転送は禁止される。第3に、補助レジスタ3の内容I2をアキュムレータ4に転送する。この転送は、補助レジスタにアドレスを付けておけば可能なことは容易に理解されよう。第4に、アキュムレータ4の内容I3をコントロールメモリ8の

(12)

内容は変化しない。第2に、アキュムレータ4の内容I2を補助レジスタ3に転送する。第3に、コントロールメモリ8のB番地のデータD1をアキュムレータ4に転送する。これにより、コントロールメモリ8のデータD1および命令ビットデータI1がアキュムレータ4および補助レジスタ3に転送されたことになる。第4に、アキュムレータ4の内容D2をメモリ2に転送すると、補助レジスタ3の内容I3は、命令ビット1に自動的に書き込まれる。これにより、コントロールメモリ8の内容D1およびI1がメモリ2および命令ビット1に書き込まれたことになる。以上説明したように、プログラマ側からは、計算機の語長が1ビット拡張されたように見える。

次に本発明の一実施例を第12図、第13図により説明する。本発明においては、補助メモリ5と補助レジスタ6が、第10図、第11図の従来方式にさらに追加設置される。補助メモリ5はビット演算を実行した場合の演算結果を格納するメモリ、補助レジスタ6は補助メモリ5に対応して

(14)

設けられるレジスタである。この構成でプログラム及び演算結果は次のようにモニタされる。

第1にメモリ2の内容D1がアキュムレータ4に転送される。この時自動的に命令ビット1の内容I1及び補助メモリ5内の演算結果A1が補助レジスタ3及び補助レジスタ6にそれぞれ転送される。第2にアキュムレータ4の内容D2をコントロールメモリ8のアドレスBに転送する。この時、補助レジスタ3及び6からコントロールメモリ8へのデータ転送は禁止される。第3に補助レジスタ3の内容I2及び補助レジスタ6の内容A2をアキュムレータ4に転送する。第4にアキュムレータ4の内容I3及びA3をコントロールメモリ8のアドレスB+1に転送する。以上の手順を経て、メモリ2、命令ビット1及び補助メモリ5の内容をコントロールメモリ8に格納できる。以後は、B番地の内容D3、B+1番地の内容I4からメモリ2の内容を解読でき、さらにB+1番地の内容A4から演算結果A1も取込むことができる。なお、命令ビットI1~I4は同一デ

(15)

プログラムメモリ内容と補助メモリ内容と、2度アクセスする必要があり、これは、計算機本体の演算を2度停止させることを意味し、高速の計算機には不適である。

以上説明したように、本発明によれば、命令語長拡張方式の計算機に、演算結果を記憶しておく補助メモリと、命令語拡張部に演算結果を取込む補助レジスタとを付加するのみで、モニタのためのプログラムステップ数の増大を防止し、計算機本体の停止時間を最小限にすることができ、従つて、計算機の性能を低下させることなくモニタ機能を向上することができる。

図面の簡単な説明

第1図は制御シーケンスの一例を示す図、第2図は第1図のシーケンスのプログラム例を示す図、第3図はCRT表示を考慮したシーケンス例を示す図、第4図は第3図のシーケンスのプログラム例を示す図、第5図は計算機のプログラムの説明図、第6図は命令の表示例を示す図、第7図は命令語を拡張した場合のメモリ構成図、第8図は命

(17)

ータで、サフィックス1~4はデータの移動順序を示す。演算結果A1~A4も同様である。また(1)~(4)は転送順序を示している。

次にプログラマから転送されてきたデータをメモリ2に格込む場合であるが、第13図に示すように、補助メモリ5及び補助レジスタ6への格込みは常に禁止しておくことにより、補助メモリ5の内容を保存することができる。

上記した実施例方法によれば、前述の命令語拡張方式の計算機に、演算結果を記憶しておく補助メモリ5と、命令語拡張部に演算結果を取込む補助レジスタ6とを追加設置するのみで、モニタのためのプログラムステップ数を増大させる必要はなくなり、またこの補助メモリは揮発性の安価なICメモリでよく、コスト上昇もわずかで済む利点がある。

上記実施例方式の他に、演算結果を格納している補助メモリの内容を、メモリ2の内容とは別々に、2度に分けて取込む方法も考えられるが、しかしこの方法では、1回のモニタをするために、

(16)

命令語拡張の回路図、第9図は一般の計算機のブロック構成図、第10図は従来の命令語読込みの説明図、第11図は従来の命令語書込みの説明図、第12図は本発明による命令語読込みの説明図、第13図は本発明による命令語書込みの説明図である。

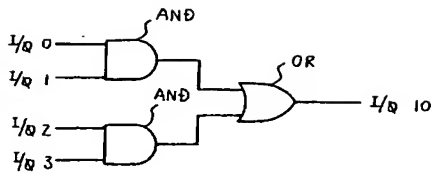
1…命令ビット、2…メモリ、3、6…補助レジスタ、4…アキュムレータ、5…補助メモリ、7、10…データバス、8…コントロールメモリ、9…専用ライン、11…プログラマ、12…処理装置、13…プログラマ制御回路。

代理人 弁理士 高橋明夫



(18)

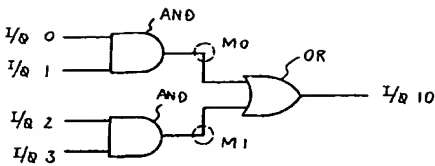
第 1 図



第 2 図

演算ステップ	プログラム	演 算
1	LD I/Q 0	$A \leftarrow (I/Q 0)$
2	AND I/Q 1	$A \leftarrow (A) \cap (I/Q 1)$
3	LD I/Q 2	$STACK \leftarrow (A) - 1$ $A \leftarrow (I/Q 2)$
4	AND I/Q 3	$A \leftarrow (A) \cap (I/Q 3)$
5	[OR]	$A \leftarrow (A) \cup (STACK)$
6	OUT I/Q 10	$I/Q 10 \leftarrow (A)$

第 3 図



第 4 図

演算ステップ	プログラム	演 算
1	LD I/Q 0	$A \leftarrow (I/Q 0)$
2	AND I/Q 1	$A \leftarrow (A) \cap (I/Q 1)$
3	OUT M 0	$M0 \leftarrow (A)$
4	LD I/Q 2	$A \leftarrow (I/Q 2)$
5	AND I/Q 3	$A \leftarrow (A) \cap (I/Q 3)$
6	OUT M 1	$M1 \leftarrow (A)$
7	OR M 0	$A \leftarrow (A) \cup (M0)$
8	OUT I/Q 10	$I/Q 10 \leftarrow (A)$

第 5 図

アドレス	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A+0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																

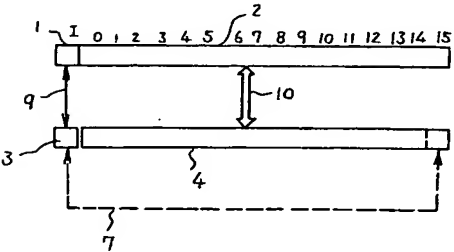
第 7 図

アドレス	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A+0																
1																
2																
3																
4																
5																
6																
7																
8																

第 6 図

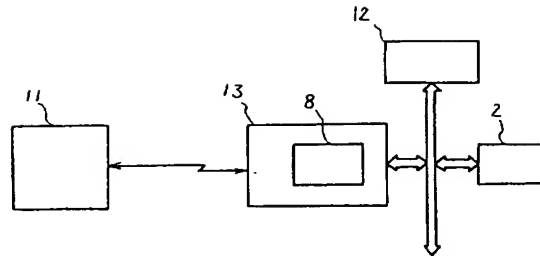
(1) 2 進表示	1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1
(2) 16 進表示	F 0 0 1
(3) 記号表示	LD 10

第 8 図

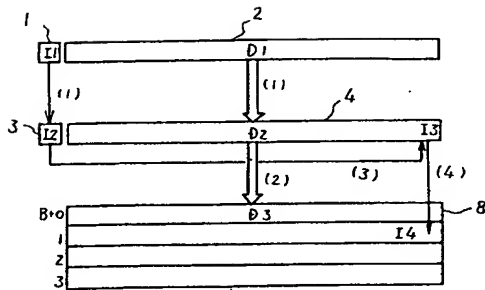




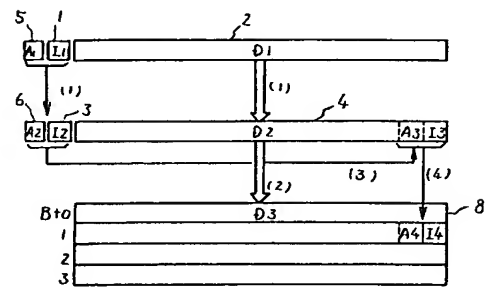
第 9 図



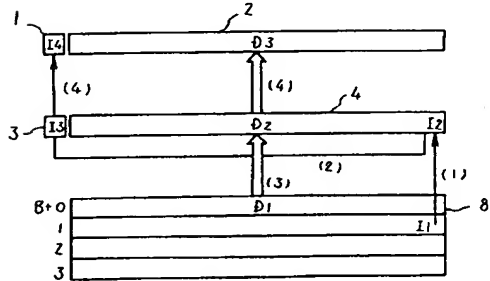
第 10 図



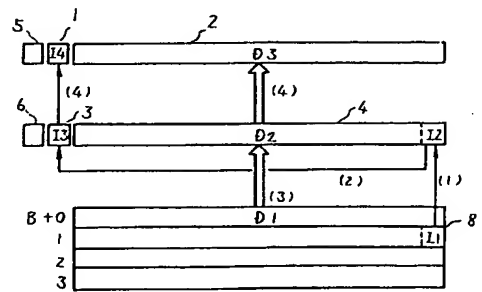
第 12 図



第 11 図



第 13 図



**THIS PAGE BLANK** (USPTO)

DOCKET NO: P 2001, 0304

SERIAL NO: 10/694,591

APPLICANT: Hartlieb et al.

LERNER AND GREENBERG P.A.

P.O. BOX 2480

HOLLYWOOD, FLORIDA 33022

TEL. (954) 925-1100